# Building the Robotium Recorder and EmitRobotiumCode

## Requirements

* Eclipse development environment
* Android SDK (samples are used in this document)
* Android ADT tools for eclipse
* Git
* Robotium!

Create a directory, initialize the project, and clone the sources

mkdir c:\doctest

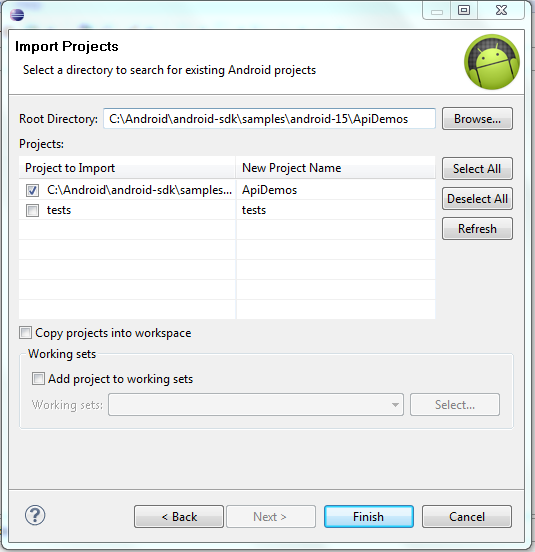
cd c:\doctest

git init .

git clone <https://github.com/mattrey555/RobotiumRecorder.git>

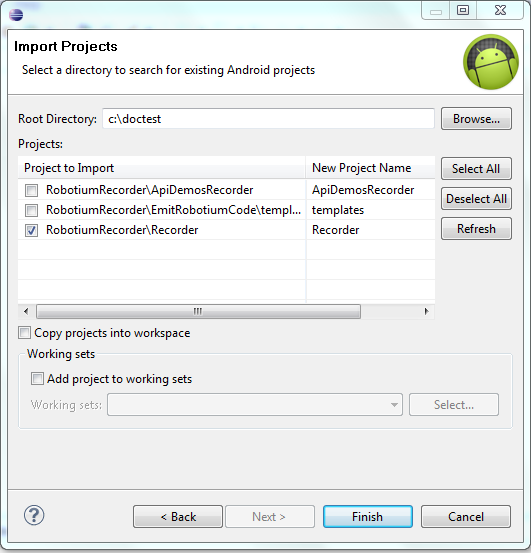
## Import the Recorder

Fire up eclipse and point it to the directory that you cloned the project into. The first task is to import the project that you’re going to record, in this case, *ApiDemos* from the Android samples:



Hit *Finish*, and the *ApiDemos* project should be imported into your workspace.

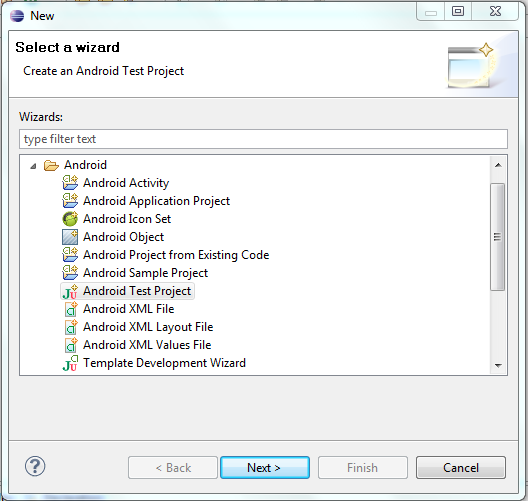
Import the recorder project, which is an Android library



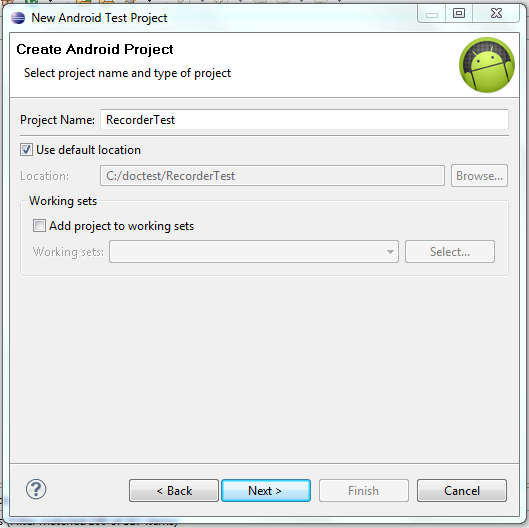
Hit *Finish*

## Create the Recording Test Project

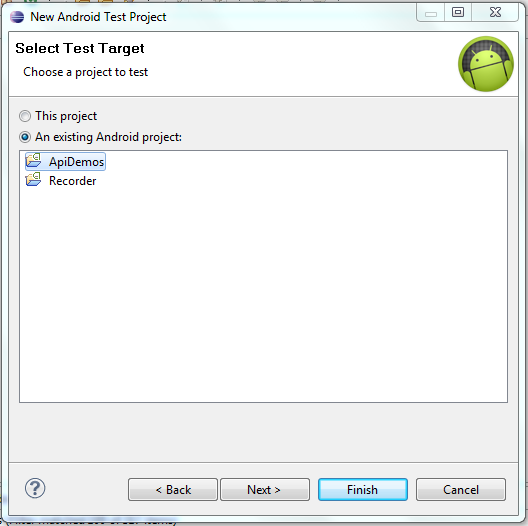
Create a new Android Test Project:



Click *Next* >



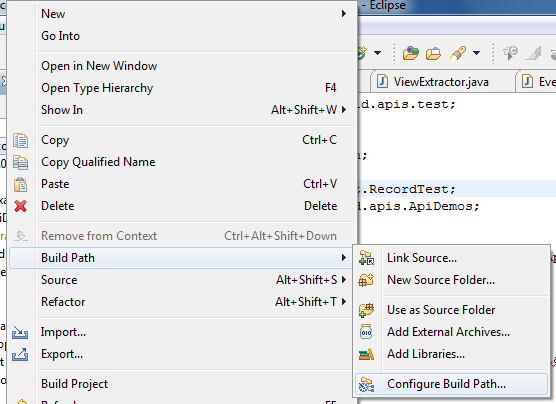
Fill in the test project name, then click *Next >*



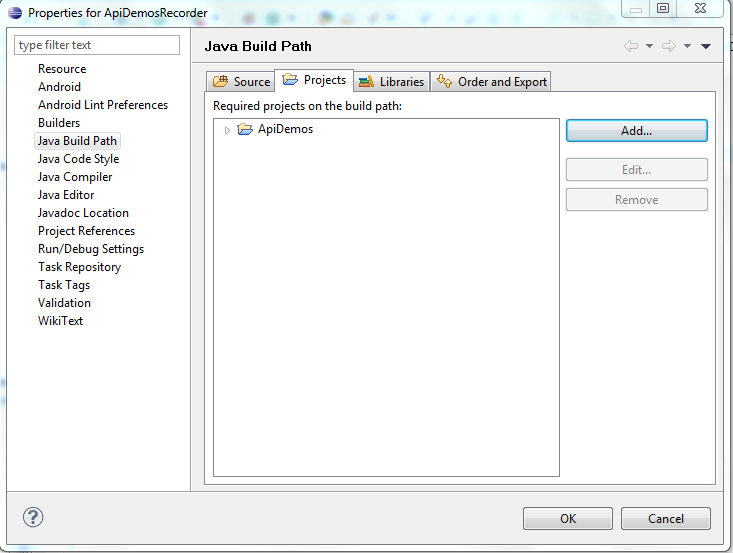
Select the project you want to record, in this case *ApiDemos*, then click *Finish*.

## Import the Recorder Library into your project

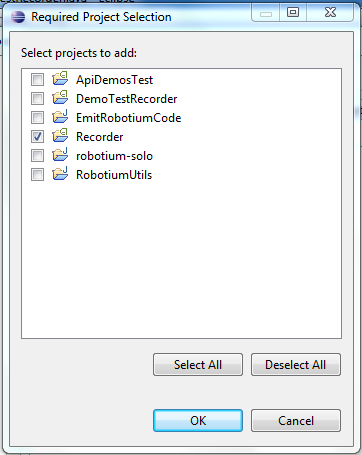
Right click on your new project, and select *configure build path:*



Then, select the *projects* tab:

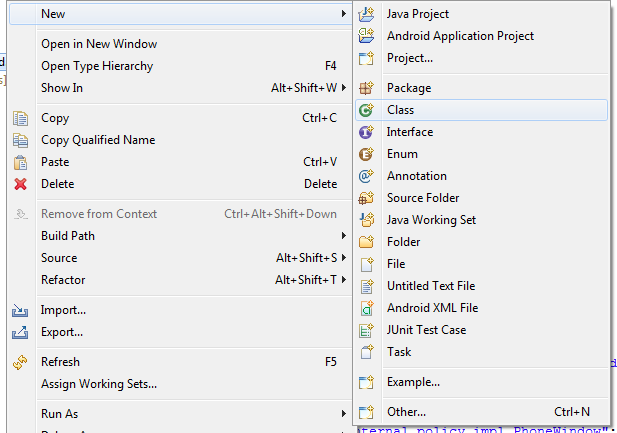


And click *Add…* A dialog will appear with the available projects in your eclipse workspace. Select *Recorder:*

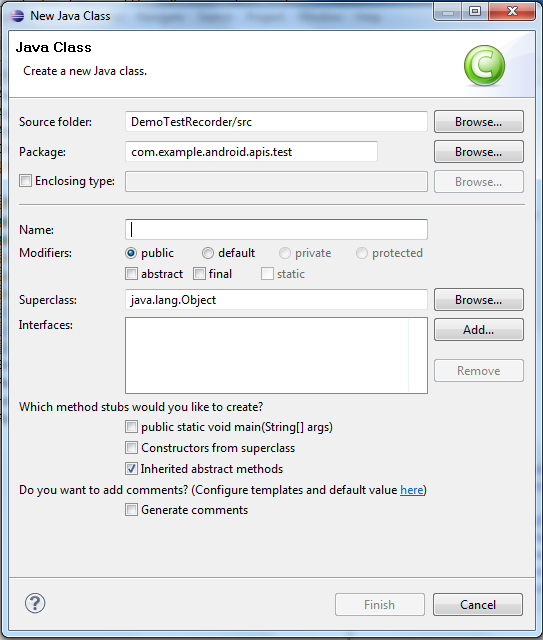


## Create the Recorder Test Class

Create a single class in the project:



Name the project *DemoTestRecorder*



The new file needs to contain the following code:

**package** com.example.android.apis.test;

**import** java.io.IOException;

// Import this to get the class for the recorder

**import** com.androidApp.Test.RecordTest;

// import the start activity of the project under test

**import** com.example.android.apis.ApiDemos;

// RecordTest is the template class for the recorder, The target class “ApiDemos” is the name of

// the application that you want to record

**public** **class** DemoTestRecorder **extends** RecordTest<ApiDemos> {

// These two constructors are required.

**public** DemoTestRecorder() **throws** IOException {

**super**(ApiDemos.**class**);

}

**public** DemoTestRecorder(Class<ApiDemos> activityClass) **throws** IOException {

**super**(activityClass);

}

// In the target project, get the class used for the R.java file in the “gen” directory,

// Then use the fully qualified path to reference the R.id class and the R.sting class.

// this provides the references for view ID and string resources, so the recorder will

// generate resource references, rather than hex values.

@Override

**public** **void** initializeResources() {

addRdotID(**new** com.example.android.apis.R.id());

addRdotString(**new** com.example.android.apis.R.string());

}

// Standard automation test setUp() function

**public** **void** setUp() **throws** Exception {

**super**.setUp();

}

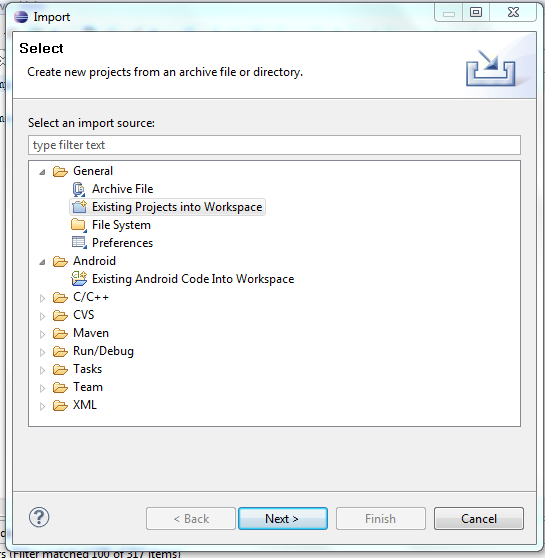
}

Compile and run the project on your device.

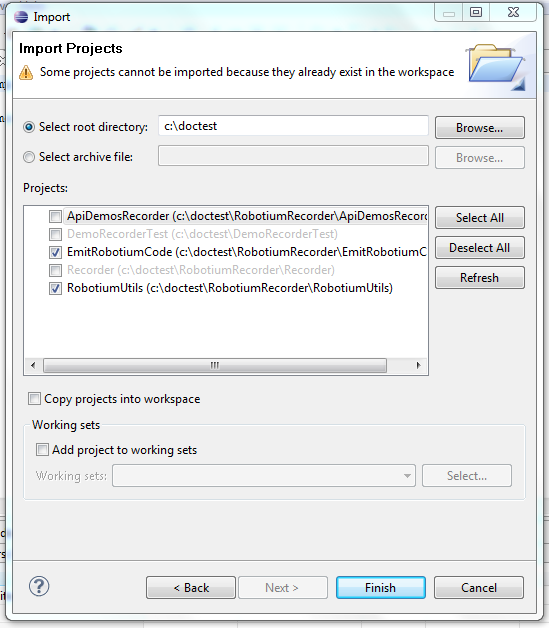
## Build EmitRobotiumCode

Now you’re ready to record. Of course, you need to generate Robotium code. This requires two projects, *EmitRobotiumCode*, which takes the recorded events from the application being recorder, and *RobotiumUtils*, a jar file which is linked with the playback project.

Import the projects by right clicking in the project explorer, and selecting *import*:

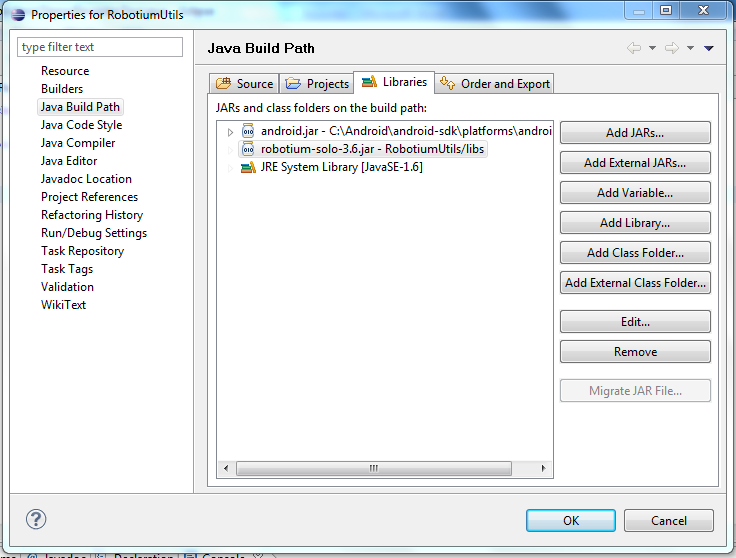


Click *Next>* then select the *EmitRobotiumCode* and *RobotiumUtils* projects:

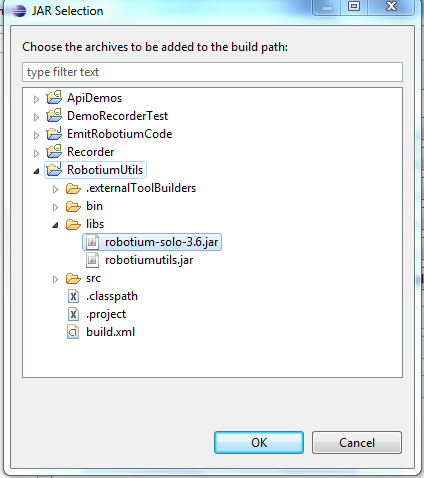


Click *Finish.*

The *RobitumUtils* project needs to have a reference to the Robotium jar file. Create a *libs* folder in the *RobotiumUtils* project and copy the robotium jar file into Right click on the *RobotiumUtils* project, and select *Configure Build Path*, and select the *libraries* tab



Pick *Add JARs…,* then select the directory that you copied the robotium jar file into:



Build the *robotiumUtils* jar file, then copy the resulting file into the EmitRobotiumCode templates directory. Just to be sure, refresh both projects before doing this.

## Generate the Robotium Test Project

When EmitRobotiumCode generates the test project, the jar file is copied into its *libs* directory, and added to the class path. You’re ready to start recording. Right click the new recorder project in the project explorer, and pick *run as Android JUnit Test*. Run through the application and exit. The recording is written to /sdcard/events.txt, so you need to have external storage on your device for the recorder to work. To examine the recording file, use *adb pull:*

adb pull /sdcard/events.txt

The output file looks like this.

activity\_forward:49461246,com.example.android.apis.ApiDemos,com.example.android.apis.ApiDemos@41248168

package:49461674,com.example.android.apis

scroll:49465445,0,11,11,class\_index,android.widget.ListView,0,1st ListView

scroll:49465460,0,11,11,class\_index,android.widget.ListView,0,1st ListView

scroll:49465468,0,11,11,class\_index,android.widget.ListView,0,1st ListView

item\_click:49466384,10,class\_index,android.widget.ListView,0,Views

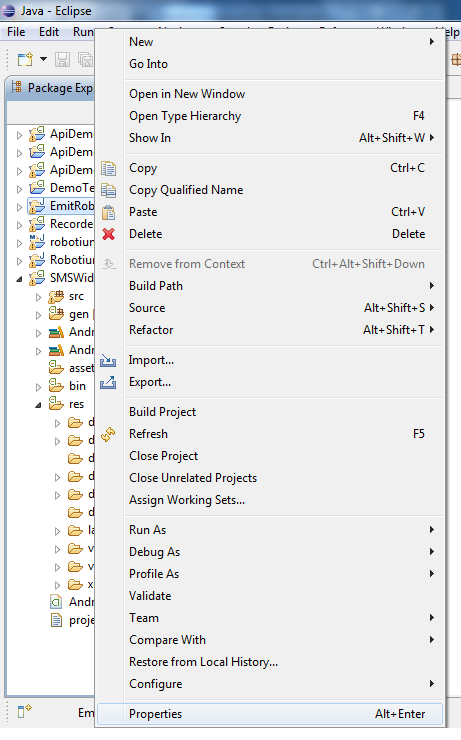
activity\_forward:49466818,com.example.android.apis.ApiDemos,com.example.android.apis.ApiDemos@41569c38

item\_click:49468824,0,class\_index,android.widget.ListView,0,Animation

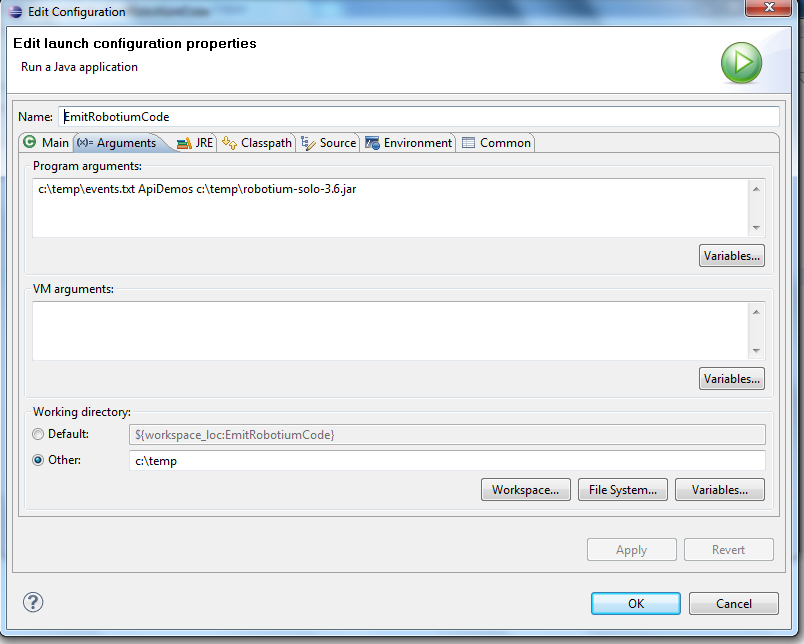
activity\_forward:49469348,com.example.android.apis.ApiDemos,com.example.android.apis.ApiDemos@41613478

item\_click:49471958,0,class\_index,android.widget.ListView,0,3D Transition

*EmitRobotiumCode* is a java program which can be run from the command line. It takes a set of parameters which specify the source events file (or *device*), the name of the project under test, and the path to the robotium jar file. In the Package Explorer, select the *EmitRobotiumCode* project, then right click for the context menu, and select *properties.*

**

In the Properties dialog, on the left hand side, there's a tree view. Select *Run/Debug Settings* from the tree view. You'll need to create new Run/Debug settings for *EmitRobotiumCode*. Click *New* then select *Java* *Application* from the popup. A tabbed dialog will appear. Select the *Arguments* tab.



In the Program Arguments field, enter *device* <*the name of the target you're testing*> <*the path to the robotium jar file on your machine*>. In the *working directory* panel, select *Other*, and enter the path that you want the test project output to. For example, to generate a test project for *ApiDemos* in c:\temp, with your robotium jar file in c:\robotium, the command line would be:

device ApiDemos c:\robotium\robotium-4.0.jar

Note: device is the actual word “device”, not the device id returned by adb.

And the working directory would be *c:\temp*

Once you've generated the project, you can import it into eclipse as an existing Android Project.

If you make successive runs of EmitRobotiumCode, it will generate new java files into the project you generated, and will ensure that they are unique by adding a numeric suffix to them, such as *ApiDemosTest1.java*, *ApiDemosTest2.java*